In the Claims:

Please amend Claims 1, 2, 5, 9, 12, 13, 16 and 20; cancel Claims 29-30; and add Claims 31-37, all as shown below.

1. (Currently Amended) A system for high availability clustering of a group of computer nodes, comprising:

one or more computers interconnected to create a cluster network, each computer including a software cluster server, a cluster database, and a set of resources of multiple resource types, including software application servers, wherein each software cluster server operating at one of the one or more of the computers provides an application access to the set of resources on said computer, or at another one of the one or more computers interconnected to the cluster network;

wherein one of the one or more computers is designated as a group leader, and the other computers are designated as members within the cluster, and wherein a cluster configuration file is maintained by the group leader to manage configuration information about the cluster, including the set of resources on each one of the one or more computers;

~~a Java-based cluster server executing on a Java virtual machine on a computer, wherein said Java-based cluster server provides an application access to a set of resources of multiple resource types , wherein two or more resource types correspond to two or more different application servers within a cluster, wherein said resources, and application servers are available at one or more computers in the cluster, and wherein the resources, and application servers are grouped by resource type within the set of resources;~~

a resource interface provided by said ~~Java-based~~ software cluster server that provides an abstraction layer [[and]] that allows the ~~Java-based~~ software cluster server to receive ~~uniform~~ requests from the application and communicate the requests to said set of resources;

a plurality of plugins that are plugged into the resource interface to provide a set of application-specific callbacks from the ~~Java-based~~ software cluster server to the set of resources, which application-specific callbacks facilitate communication of the requests from the application to the set of resources, wherein the resource interface accepts additional plugins that are plugged into the resource interface to provide application-specific callbacks from the software cluster servers to other resource types; ~~the system includes a plugin for each resource type corresponding to the different application server, and wherein each plugin implements a resource API to encapsulate the plugin's particular resource type-specific behavior and to isolate the Java-based cluster server from said behavior while providing access to its pool of resources; and~~

~~a JNDI interface provided by said cluster server, wherein the JNDI interface provides an interface between the Java-based cluster server and a JNDI-compliant database;~~

~~wherein the resource interface accepts additional plugins that are plugged into the resource interface for other resource types; and~~

wherein each computer in the cluster communicates the set of resources available on said computer to the group leader, and wherein when the requests from the application are received, the group leader

determines the availability of the set of resources on each one of the one or more computers by referencing the cluster configuration file, and

directs the request to the computer having the requested resource;

wherein the system can be extended by adding additional computers with cluster servers and resource interfaces operating thereon.

~~a GLobal Update Protocol (GLUP) mechanism that employs a distributed global lock with sequence numbers to serialize propagation of global events across active members of the cluster, wherein each global update is associated with a unique sequence number such that each said active member of the cluster has an identical view of an ordering of the global events.~~

2. (Currently Amended) The system of claim 1 wherein each of said ~~Java-based~~ software cluster servers includes a heartbeat interface that provides heartbeat information to other ~~Java-based~~ software cluster servers at said other application servers.

3. – 4. (Canceled)

5. (Currently Amended) The system of claim 1 wherein the system includes a cluster administration utility for accessing and administering the ~~Java-based~~ software cluster server using remote method invocation calls.

6. (Original) The system of claim 1 wherein each resource has a resource type associated with it.

7. (Original) The system of claim 6 wherein resources are the object instances of their respective resource types.

8. (Original) The system of claim 1 wherein a resource is any of a computer, internet protocol address, disk, database, or file system or application.

9. (Currently Amended) The system of claim 1 wherein the ~~Java-based~~ software cluster server defines resource groups that includes clusters of resources.

10. – 11. (Canceled)

12. (Currently Amended) A method for providing a high availability clustering framework system for a group of computer nodes, comprising the steps of:

allowing an ~~software~~ application to access, via a ~~Java-based~~ software cluster server ~~executing on a Java virtual machine on a computer~~, a set of resources of various resource types, located on one or more computers interconnected to create a cluster network, ~~wherein two or more resource types correspond to two or more different application servers within a Java-based cluster wherein said resources are available at said computer or at another computer, and wherein the resources and application servers are grouped by resource type within the set of resources;~~

wherein each computer includes a software cluster server, a cluster database, and the set of resources of multiple resource types, including software application servers;

wherein one of the one or more computers is designated as a group leader, and the other computers are designated as members within the cluster, and wherein a cluster configuration file is maintained by the group leader to manage configuration information about the cluster, including the set of resources on each one of the one or more computers;

providing a resource interface at said ~~Java-based~~ software cluster server that provides an abstraction layer [[and]] that allows the ~~Java-based~~ software cluster server to receive ~~uniform~~ requests from the ~~Java~~ application and communicate the requests to said set of resources via a plurality of plugins that are plugged into the resource interface;

wherein the plurality of plugins are plugged into the resource interface to provide a set of application-specific callbacks from the ~~Java-based~~ software cluster server to the set of resources, which application-specific callbacks facilitate communication of the requests from the application to the set of resources, wherein the resource interface accepts additional plugins that are plugged into the resource interface to provide application-specific callbacks from the software cluster servers to other resource types; ~~the system includes a plugin for each resource type corresponding to the different application server, and wherein each plugin implements a resource API to encapsulate the~~

~~plugin's particular resource type-specific behavior and to isolate the Java-based cluster server from said behavior while providing access to its pool of resources~~

~~wherein a JNDI interface provides an interface between the Java-based cluster server and a JNDI-compliant database;~~

~~wherein the resource interface accepts additional plugins that are included in the resource interface for other resource types;~~

wherein each computer in the cluster communicates the set of resources available on said computer to the group leader, and wherein when the requests from the application are received, the group leader

determines the availability of the set of resources on each one of the one or more computers by referencing the cluster configuration file, and

directs the request to the computer having the requested resource;

wherein the system can be extended by adding additional computers with cluster servers and resource interfaces operating thereon.

~~providing a GLobal Update Protocol (GLUP) mechanism that employs a distributed global lock with sequence numbers to serialize propagation of global events across active members of the cluster, wherein each global update is associated with a unique sequence number such that each said active member of the cluster has an identical view of an ordering of the global events.~~

13. (Currently Amended) The method of claim 12 wherein said ~~Java-based~~ software cluster server includes a heartbeat interface provides heartbeat information to other ~~Java-based~~ software cluster servers at said other application servers.

14. – 15. (Canceled)

16. (Currently Amended) The method of claim 12 wherein the system includes a cluster administration utility for accessing and administering the ~~Java-based~~ software cluster server using remote method invocation calls.

17. (Original) The method of claim 12 wherein each resources has a resource type associated with it.

18. (Original) The method of claim 17 wherein resources are the object instances of their respective resource types.

19. (Original) The method of claim 12 wherein a resource is any of a computer, ip address, disk, database, or file system or application.

20. (Currently Amended) The method of claim 12 wherein the ~~Java-based~~ software cluster server allows for clustering resources within a resource group.

21. – 30. (Canceled)

31. (New) A computer-readable storage medium carrying one or more sequences of instructions, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

allowing an application to access, via a software cluster server, a set of resources of various resource types, located on one or more computers interconnected to create a cluster network,

wherein each computer includes a software cluster server, a cluster database, and the set of resources of multiple resource types, including software application servers;

wherein one of the one or more computers is designated as a group leader, and the other computers are designated as members within the cluster, and wherein a cluster configuration file is maintained by the group leader to manage configuration information about the cluster, including the set of resources on each one of the one or more computers;

providing a resource interface at said software cluster server that provides an abstraction layer that allows the software cluster server to receive requests from the application and communicate the requests to said set of resources via a plurality of plugins that are plugged into the resource interface;

wherein the plurality of plugins are plugged into the resource interface to provide a set of application-specific callbacks from the software cluster server to the set of resources, which application-specific callbacks facilitate communication of the requests from the application to the set of resources, wherein the resource interface accepts additional plugins that are plugged into the resource interface to provide application-specific callbacks from the software cluster servers to other resource types;

wherein each computer in the cluster communicates the set of resources available on said computer to the group leader, and wherein when the requests from the application are received, the group leader

determines the availability of the set of resources on each one of the one or more computers by referencing the cluster configuration file, and

directs the request to the computer having the requested resource;

wherein the system can be extended by adding additional computers with cluster servers and resource interfaces operating thereon.

32. (New) The method of claim 31, wherein the resource interface provides an interface between the software cluster server and a database, and wherein the resource interface accepts additional plugins that are plugged into the resource interface to provide application-specific callbacks from the software cluster servers to other resource types.

33. (New) The method of claim 31, further comprising:

providing a plugin for each resource type corresponding to a different application server, and wherein each plugin implements a resource API to encapsulate the plugin's particular resource type-specific behavior, and to isolate the software cluster server from said behavior while providing access to each application server's set of resources.

34. (New) The system of claim 1, wherein the resource interface provides an interface between the software cluster server and a database, and wherein the resource interface accepts additional plugins that are plugged into the resource interface to provide application-specific callbacks from the software cluster servers to other resource types.

35. (New) The method of claim 12, wherein the resource interface provides an interface between the software cluster server and a database, and wherein the resource interface accepts additional plugins that are plugged into the resource interface to provide application-specific callbacks from the software cluster servers to other resource types.

36 (New) The system of claim 1, including a plugin for each resource type corresponding to a different application server, and wherein each plugin implements a resource API to encapsulate the plugin's particular resource type-specific behavior, and to isolate the software cluster server from said behavior while providing access to each application server's set of resources.

37. (New) The method of claim 12, further comprising:

providing a plugin for each resource type corresponding to a different application server, and wherein each plugin implements a resource API to encapsulate the plugin's particular resource type-specific behavior, and to isolate the software cluster server from said behavior while providing access to each application server's set of resources.